**INTRODUCCIÓN**

En el mundo del desarrollo de software, los procesos ágiles han revolucionado la forma en que se crean productos. A diferencia de los métodos tradicionales basados en cascadas, los procesos ágiles se caracterizan por su flexibilidad, adaptabilidad y enfoque en la colaboración.

Los procesos ágiles son metodologías de desarrollo de software que se basan en ciclos cortos de trabajo, conocidos como sprints. Durante cada sprint, un equipo de desarrollo trabaja en conjunto para completar una serie de funcionalidades definidas previamente. Al finalizar el sprint, se presenta el trabajo realizado al cliente y se recolecta retroalimentación para mejorar en el siguiente ciclo.

El Manifiesto Ágil es un documento que se publicó en el año 2001 y que resume los valores y principios fundamentales de los procesos ágiles. Este manifiesto está basado en cuatro pilares:

* **Individuos e interacciones sobre procesos y herramientas**
* **Software funcionando sobre documentación extensa**
* **Colaboración con el cliente sobre negociación contractual**
* **Respuesta ante el cambio sobre seguir un plan**

En el mundo del desarrollo de software moderno, la asociación entre las metodologías ágiles y la Integración y Entrega Continua (CI/CD) se ha convertido en una fórmula ganadora para optimizar procesos y mejorar la calidad del software.

**¿Cómo se complementan las metodologías ágiles y CI/CD?**

Las metodologías ágiles, como Scrum o Kanban, se caracterizan por dividir el trabajo en ciclos cortos o sprints, donde se prioriza la colaboración, la retroalimentación constante y la adaptación a los cambios. Por su parte, CI/CD automatiza las tareas de integración y entrega de software, asegurando que el código siempre esté actualizado, probado y listo para ser implementado en producción.

**UNIDAD 1: METODOLOGIAS AGILES**

En el mundo del desarrollo de software, las metodologías ágiles se han convertido en una alternativa popular a los enfoques tradicionales basados en cascadas. Entre estas metodologías ágiles, Scrum destaca por su simplicidad y efectividad.

**Y, ¿Qué son las metodologías ágiles?**

Las metodologías ágiles se basan en la idea de dividir el trabajo en pequeños ciclos o sprints, con el objetivo de obtener retroalimentación constante del cliente y adaptar el proyecto en consecuencia. Esto permite una mayor flexibilidad y una mejor respuesta a los cambios en las necesidades del cliente.

**Pero, ¿Qué es Scrum?**

Scrum es una metodología ágil específica que utiliza un conjunto de roles, eventos y artefactos para gestionar el desarrollo de software. Los roles principales en Scrum son el Product Owner, el Scrum Master y el equipo de desarrollo.

**Y ¿Cómo funciona Scrum?**

En Scrum, el trabajo se divide en sprints, que suelen durar entre una y cuatro semanas. Al comienzo de cada sprint, el Product Owner selecciona las funcionalidades que se deben desarrollar en ese sprint. El equipo de desarrollo luego estima el tiempo que llevará completar cada funcionalidad.

Durante el sprint, el equipo de desarrollo trabaja de forma autónoma para completar las funcionalidades seleccionadas. El Scrum Master facilita el trabajo del equipo y elimina cualquier obstáculo que pueda impedir su progreso.

Al final de cada sprint, el equipo de desarrollo presenta las funcionalidades completadas al Product Owner y a las partes interesadas. El Product Owner proporciona retroalimentación sobre el trabajo y selecciona las funcionalidades que se incluirán en la próxima versión del producto.

**Algunos Beneficios de las metodologías ágiles y Scrum**

Las metodologías ágiles y Scrum ofrecen una serie de beneficios, como:

* **Mayor flexibilidad y capacidad de adaptación a los cambios**
* **Mejor comunicación y colaboración entre el equipo de desarrollo y el cliente**
* **Entrega más rápida de software de alta calidad**
* **Mayor satisfacción del cliente**

**Conclusión**

Las metodologías ágiles y Scrum son herramientas valiosas para el desarrollo de software exitoso en el mundo actual, que se caracteriza por la rapidez de los cambios y la necesidad de una respuesta flexible a las necesidades del cliente.

**UNIDAD 2: INTEGRACION Y ENTREGA CONTINUA**

En el mundo del desarrollo de software moderno, la Integración y Entrega Continua (CI/CD) se ha convertido en una práctica esencial para optimizar los procesos de desarrollo y lanzamiento de software. Esta metodología permite automatizar las tareas repetitivas y tediosas, liberando tiempo para que los desarrolladores se enfoquen en actividades más creativas y de alto valor.

**Y, ¿Qué es la Integración Continua (CI)?**

La Integración Continua (CI) es una práctica que consiste en automatizar la integración de los cambios realizados en el código fuente de un proyecto por parte de diferentes desarrolladores. Esto implica que cada vez que un desarrollador realiza un cambio, el código se integra automáticamente en un repositorio central, donde se ejecutan pruebas automatizadas para detectar errores o problemas.

**Y ¿Qué es la Entrega Continua (CD)?**

La Entrega Continua (CD) va un paso más allá de la CI al automatizar el proceso de entrega de software a producción. Una vez que las pruebas automatizadas en CI se ejecutan con éxito, CD se encarga de empaquetar el código, implementarlo en el entorno de producción y realizar pruebas finales para garantizar que el software funciona correctamente.

**Algunos Beneficios de la Integración y Entrega Continua (CI/CD)**

Los beneficios de implementar CI/CD en los proyectos de software son numerosos, entre los que destacan:

* **Mayor velocidad de entrega de software:** Al automatizar las tareas manuales, CI/CD permite reducir significativamente el tiempo que se tarda en lanzar nuevas versiones de software.
* **Mejor calidad del software:** Las pruebas automatizadas que se ejecutan en CI/CD ayudan a detectar errores y problemas de forma temprana, lo que reduce la cantidad de errores en el software final.
* **Mayor confiabilidad del software:** CI/CD garantiza que el software que se implementa en producción sea siempre la última versión y que haya pasado todas las pruebas necesarias.
* **Mayor satisfacción del cliente:** Al entregar software de alta calidad con mayor frecuencia, las empresas pueden mejorar la satisfacción de sus clientes.

**Podemos utilizar herramientas para CI/CD como:**

Existen numerosas herramientas disponibles para implementar CI/CD, como Azure DevOps, Jenkins, GitLab CI/CD, Travis CI y CircleCI. Cada herramienta ofrece diferentes características y funcionalidades, por lo que es importante elegir la que mejor se adapte a las necesidades específicas de cada proyecto.

**Conclusión**

La Integración y Entrega Continua (CI/CD) se ha convertido en una práctica fundamental para el desarrollo de software moderno. Al automatizar las tareas repetitivas y tediosas, CI/CD permite a los equipos de desarrollo trabajar de manera más eficiente, entregar software de alta calidad con mayor frecuencia y mejorar la satisfacción del cliente.